

Davide BIGGIN

Socio UAI  
dabiggi@alice.it

# Una proposta di progetto per il *Database* UAI

## **Abstract**

*Within the project "Database UAI", this first work focuses on the logical and conceptual design of the database. After examining main existing software, evaluated in relation to the needs of the project, we introduce a theoretical digression on models of database design. Is then, is presented and commented the Entity-Relationship preliminary schema. Below is the translation in the Relational Model. Will not be discussed in this article the implementation aspects, still under discussion.*

## **Introduzione**

La realizzazione di un *database* è un processo impegnativo sia dal punto di vista progettuale che di implementazione. Per questa ragione prima di pensare alla progettazione (e poi allo sviluppo) di una nuova piattaforma, può essere utile valutare se e in che misura i prodotti già in essere sul mercato possano soddisfare le esigenze richieste.

Queste ultime vanno quanto prima formalizzate, cercando di delineare le peculiarità che una piattaforma di organizzazione delle immagini astronomiche debba avere; proprio il Congresso UAI di Settembre 2012 sarà la giusta occasione. In quest'ottica mi propongo di passare in rassegna alcuni tra gli strumenti presenti sul mercato.

In seguito, dopo una necessaria digressione teorica sugli strumenti di progettazione di una base di dati, ne propongo il modello concettuale utilizzando i costrutti formali che si adoperano nell'Ingegneria del Software. Termina la trattazione la conversione dello schema concettuale nel modello Relazionale definitivo.

## **Le esigenze.**

Prima di iniziare qualunque argomentazione bisogna delineare quali sono le esigenze principali che il sistema deve avere.

Nell'attesa quindi del già citato Congresso di Settembre è mia intenzione provare a delinearne alcune sulla base della mia esperienza sia nel campo dello sviluppo del software che di astrofilo dilettante:

*Accesso via web:* E' chiaro che il sistema dovrà essere accessibile via web tramite i comuni *browser* e compatibile quindi con tutti i sistemi operativi esistenti (Windows, MAC OS X, Linux, ecc);

*Gestione della multiutenza:* Si suppone che sia previsto un sistema di autenticazione dell'utente che opera sul sistema;

*Supporto a diversi formati immagine:* Trattandosi di un archivio di immagini astronomiche, dovranno essere supportati formati specifici (FITS o RAW) oltre al JPEG già diffuso in altri contesti;

*Query di Ricerca:* Questo è un aspetto particolarmente importante. Occorre definire quali parametri saranno l'oggetto delle ricerche all'interno della base di dati. Per fare alcuni esempi: ricercare un'immagine in base all'oggetto ripreso, piuttosto che in base alla data, o all'autore oppure ad una combinazione di questi elementi. La struttura del modello concettuale, infatti, deve consentire l'elaborazione di *query* quanto più articolate possibili;

*Semplicità d'uso:* Occorre altresì che il sistema non sia eccessivamente macchinoso nel suo uso, sia in sede di caricamento delle immagini, che di ricerca. Come vedremo più diffusamente nel seguito, esistono complessi *database* astronomici professionali che però sono poco intuitivi e richiedono inoltre competenze informatiche notevoli per poterli utilizzare al meglio;

*Basso costo:* Sia in termini di progettazione, sviluppo ed implementazione che di manutenzione. Si consiglia quindi di optare per piattaforme *opensource* che offrono da un lato sicura affidabilità, dall'altro basso costo di acquisto.

Queste sono solo alcune delle possibili esigenze della base di dati, l'elenco potrà essere certamente integrato in sede di discussione congressuale.

### I software esistenti.

Tutti coloro che navigano spesso in rete avranno sicuramente notato quanti siti web propongono delle gallerie di fotografie, scattate sia da professionisti che da semplici amatori.

Molti di questi sono gestiti da cosiddetti CMS (*Content Management System*) che possiamo definire come degli strumenti *software*, installati su un server web, il cui compito è facilitare la gestione dei contenuti di un sito, svincolando l'amministratore da conoscenze tecniche e di programmazione. Esistono pertanto varie tipologie di CMS: dai gestionali dei blog, ai forum di discussione, ai dizionari e, naturalmente, alle *Image Gallery*. Queste ultime, consentono di pubblicare delle fotografie, suddividendole in album, con in più la possibilità di inserire dei "tag". Il tag ha lo scopo di identificare (o meglio "referenziare") l'immagine stessa al fine di renderne più efficace l'identificazione all'interno di una ricerca. Cito alcuni dei CMS che ho preso in considerazione: *Coppermine*<sup>1</sup>, *GalleryCMS*<sup>2</sup>, e *Piwigo*<sup>3</sup>. Quest'ultimo merita, a mio avviso, una menzione particolare perché, tra quelli testati, è risultato il più completo e funzionale. Infatti tra le sue caratteristiche, la possibilità di organizzare le immagini in Categorie ad albero senza limite di profondità, e referenziarle con i tag. I requisiti software necessari per il funzionamento del sistema sono assolutamente in linea con quelli richiesti dalla maggior parte dei CMS *OpenSource*, ovvero un server web, un *server database* MySQL e l'interprete al linguaggio di programmazione PHP; ogni provider commerciale, ma anche qualunque realtà universitaria che fornisce servizi di *hosting* è in grado di provvedere a queste necessità. Occorre però precisare che *Piwigo* è pensato e progettato per un pubblico il più vasto possibile: dal fotografo professionista che vuole pubblicizzare i propri lavori, al turista che desidera mostrare al mondo le fotografie scattate durante i suoi viaggi. Attenzione ad una distinzione importante: un conto è pubblicare una immagine astronomica sul web, a bassa risoluzione, in formato JPEG, e di un soggetto facilmente riconoscibile (un pianeta, una cometa o un oggetto di *deep sky*); un altro è catalogare una fotografia astronomica che abbia però valore scientifico. In questa circostanza sono altrettanto importanti attributi quali: data ed ora in T.U., luogo di osservazione, caratteristiche del telescopio utilizzato e/o dell'apparecchio fotografico, tecnica utilizzata, esposizione e molte altre; l'insieme di queste proprietà connota una specifica immagine, rendendola quindi spendibile per la comunità scientifica che può utilizzarla per fini di ricerca. Si pensi, ad

esempio, alla ricerca di esopianeti: la precisa indicazione della strumentazione adoperata e dei tempi di acquisizione dell'immagine possono fornire un utile strumento per coloro i quali intendono elaborare la curva di luce per quantificare precisamente le dimensioni di un ipotetico pianeta in orbita attorno alla stella. Tutto ciò non è previsto in *Piwigo*, neppure utilizzando forzatamente i tag, che sono pensati per classificare il contenuto dell'immagine piuttosto che informazioni accessorie. Quindi per chiarire bene ricapitoliamo i possibili vantaggi/svantaggi dell'uso di *Piwigo* nel contesto del database UAI.

### Vediamone i vantaggi:

Abbattimento del costo/tempo di progettazione ed implementazione: è sufficiente scaricare il programma, installarlo e configurarlo per avere un sito in funzione; Gestione degli accessi già implementato all'interno del CMS;

Gli **svantaggi** invece sono più significativi: Non è pensato per le esigenze della fotografia astronomica con finalità di ricerca scientifica; Non supporta formati specifici quali FITS o RAW; Non prevede criteri di ricerca delle immagini in base al soggetto ripreso;

Quindi, in definitiva, considerato che gli svantaggi sono bloccanti rispetto ai vantaggi, ritengo che l'uso di *Piwigo* non sia applicabile per le esigenze del progetto.

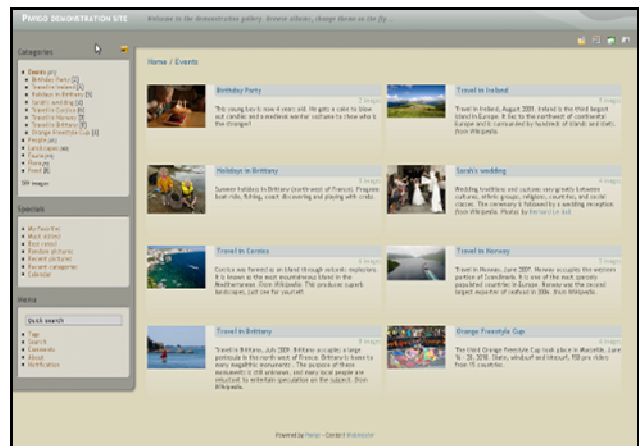


Figura 1. Uno screenshot del CMS Piwigo

### Alcuni sistemi astronomici professionali

In letteratura sono anche presenti alcune ricerche inerenti la progettazione e lo sviluppo di database astronomici fotografici professionali. Tra gli articoli più rappresentativi ho notato [1] che sebbene sia piuttosto datato (1994) contiene sicuramente degli spunti interessanti. L'obiettivo del prototipo è di realizzare una base di dati specifica per catalogare le immagini ottenute digitalizzando le lastre fotografiche del Telescopio Schmidt UK. Quello che è importante per i nostri scopi è

<sup>1</sup> <http://coppermine-gallery.net/>

<sup>2</sup> <http://www.gallerycms.com/>

<sup>3</sup> <http://it.piwigo.org/>

comprendere quali informazioni (i cosiddetti *metadata*) sono associate a ciascuna immagine: le coordinate celesti, le informazioni circa il processo di digitalizzazione (data ed ora, apertura utilizzata nel microdesintometro, *etc.*). Nel prototipo oggetto di questo studio sono stati ripresi gli Ammassi Globulari nelle Nubi di Magellano, sia con immagini a bassa risoluzione, in formato GIF per la sola consultazione *on-line*, che a risoluzione piena (FITS). La struttura del sistema è, per così dire, piuttosto classica: un *web server* attende le richieste di accesso dai *client* che vengono indirizzate ad un insieme di sottoprocessori ciascuno dei quali si occupa di gestire un aspetto specifico della richiesta: il *SQL Server* interpreta la logica della richiesta di accesso (in gergo tecnico si dice che “processa la *query*”), la passa al *DataBase Server* che recupera i *metadata*, e anche al *Data Server* che materialmente “pesca” l’immagine di interesse dall’archivio (normalmente memorizzato su *hard disk*). Un processore grafico (*Graphic Server*), infine, produce l’immagine in uscita rendendola visibile sul *browser* dell’utente. L’archivio è realizzato mediante un *database Relazionale* (per una definizione formale vedasi [2]) in cui la tabella<sup>4</sup> principale (*fieldsTAB*) contiene il nome dell’ammasso globulare, le sue coordinate celesti, i dati di acquisizione della lastra fotografica nonché del processo di digitalizzazione; da ultimo sono presenti due puntatori rispettivamente all’immagine a bassa risoluzione (GIF) e ad alta (FITS). Un aspetto senza dubbio negativo, riguarda l’interfaccia web da utilizzarsi per interrogare l’archivio: certamente non la si può considerare *user friendly* in quanto l’utente deve conoscere il linguaggio di interrogazione delle basi di dati (SQL) per poter interrogare l’archivio alla ricerca dell’immagine di suo interesse; occorre però tener presente che nell’anno in cui è stato pubblicato l’articolo (1994) i sistemi *web based* di accesso ai *database* erano ancora agli inizi. Questo lavoro resta comunque un importante spunto di riflessione per la progettazione del database UAI.

Un secondo progetto che ho preso in esame è lo *Sloan Digital Sky Survey’s (SDSS’s)* [3].

In questo articolo viene descritta la complessa architettura del sistema che conta decine di TeraBytes di dati.<sup>5</sup> I punti salienti di nostro interesse riguardano, come al solito, lo schema della base di dati.

Anche in questo caso il *database* di supporto è un Relazionale: gli autori descrivono infatti che un primo esperimento utilizzando un sistema ad oggetti (*Object*

*Oriented*: un altro modello di sviluppo molto in voga negli ultimi anni) è risultato insoddisfacente poiché le prestazioni erano insufficienti rispetto alle attese.

Le numerose tabelle che lo compongono (circa un centinaio) sono suddivise in gruppi funzionali: le tabelle con i dati fotometrici (*Photo group*), le tabelle con i dati spettroscopici (*Spectro group*), le tabelle che contengono i *metadata* (*Meta group*), e così via.

Prendere da esempio questa architettura è, a mio avviso, fuorviante poiché è progettata per gestire un archivio di dimensioni straordinariamente importanti e, non dimentichiamolo, decine di analisti e programmatori esperti hanno preso parte al progetto.

### La progettazione concettuale, cenni teorici

Abbiamo visto nei paragrafi precedenti che già esistono dei sistemi di catalogazione delle immagini astronomiche; probabilmente il più significativo è appunto lo *SDSS’s*.

Tuttavia per la sua complessità, occorre pensare ad un “nuovo” *database*, personalizzato per le esigenze della UAI, una realtà fatta di astrofili non professionisti e che non hanno le risorse sia economiche che tecniche per poter realizzare un sistema complesso come quello brevemente descritto sopra.

Ecco quindi l’esigenza di pensare ad una nuova soluzione che sia più semplice concettualmente, di facile implementazione (e questo punto è ancora aperto ad ogni discussione) e che, si spera, sia di facile utilizzo da parte della comunità. Entriamo quindi nel merito della progettazione.

Come teorizzato nei testi di Ingegneria del Software, la progettazione di una base di dati inizia con delle “interviste” che i progettisti rivolgono agli utenti/clienti del prodotto finito. Questa fase, almeno in questo momento, è stata assente, perciò ho dovuto effettuare delle scelte (del tutto opinabili, sia chiaro) in maniera arbitraria; questo porta al passaggio successivo, ovvero l’elaborazione del Modello Concettuale della base di dati, che viene formalizzato utilizzando i costrutti del cosiddetto Modello Entità-Relazione (in inglese *Entity-Relationship – E-R*) [4]. Lo schema *E-R* è stato descritto da Chen [5] nel 1976, ma ancora oggi è riconosciuto come lo strumento principale nella progettazione concettuale di una base di dati. Si tratta di un insieme di costrutti grafici (rettangoli, rombi, *etc*) simili per struttura ad un diagramma di flusso. Possiamo definire la *Progettazione Concettuale* come il passaggio da un insieme non organizzato di informazioni, di idee, di opinioni personali, di consuetudini ad un insieme di informazioni ben definite, formalizzate e comprensibili anche da chi non conosce nello specifico il prodotto finale che si desidera realizzare. Per meglio comprendere

<sup>4</sup> Nel seguito vedremo cosa si intende per “tabella” nell’ambito delle basi di dati.

<sup>5</sup> <http://skyserver.sdss.org>

il resto della trattazione è necessaria una digressione teorica che cercherò di sintetizzare il più possibile. Iniziamo descrivendo i costrutti fondamentali del modello E-R.

**Entità:** Rappresentano classi di oggetti (cose, persone, fotografie) che hanno proprietà comuni ai fini dell'applicazione di interesse. Ad esempio CITTA' è un'entità, mentre Milano, Torino sono occorrenze dell'entità CITTA'. Graficamente ogni entità viene rappresentata da un rettangolo.

**Relazioni (o Associazioni):** Rappresentano legami logici, significativi per l'applicazione di interesse, tra due o più entità. Ad esempio RESIDENZA è una relazione tra le entità CITTA' e PERSONA. Graficamente ogni relazione viene rappresentata da un rombo.

**Attributi:** Descrivono le proprietà elementari di Entità o Relazioni<sup>6</sup> che sono di interesse ai fini dell'applicazione. Ad esempio *Nome*, *Cognome* sono attributi dell'entità PERSONA. Graficamente sono rappresentati come un pallino associato all'entità (o alla Relazione) cui si riferiscono.

Occorre ancora definire il concetto della cardinalità delle relazioni. Possiamo chiarire questo punto omettendo la definizione formale (comunque reperibile in [4]) ma intuitivamente ricorrendo ad un esempio.

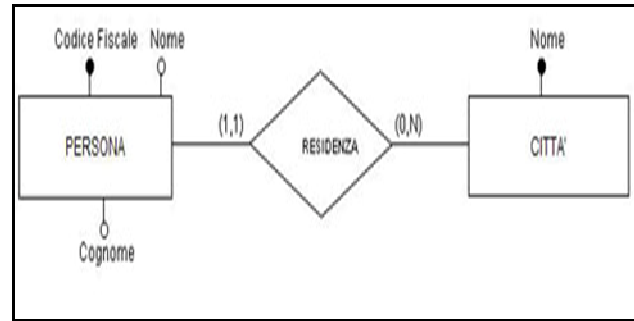
Siano PERSONA e CITTA' due entità, e ne siano *Codice Fiscale*, *Nome*, *Cognome* gli attributi<sup>7</sup>; sia RESIDENZA la relazione tra le due entità, così come raffigurato in figura 2.

I valori (1,1) e (0,N) accanto alla relazione RESIDENZA si leggono in questo modo:

Una PERSONA deve RISIEDERE in una e una sola CITTA' → una persona per una città: (1,1).

Una CITTA' può non avere nessun RESIDENTE oppure può averne un numero qualunque. → zero persone residenti oppure N persone residenti: (0,N).

Questa valorizzazione della cardinalità delle relazioni è molto importante ai nostri fini perché ci permette di stabilire dei vincoli di correttezza sui dati. Per fare un esempio calato nel contesto, possiamo voler indicare che una PERSONA è AUTORE di almeno una fotografia (altrimenti non avrebbe senso memorizzarne gli attributi all'interno del *database*), ma anche di più di una (1,N); una fotografia, d'altro canto, può avere come autore almeno una ma anche più persone (1,N). Concludiamo questa digressione teorica con una definizione molto importante:



**Figura 2.** La relazione RESIDENZA tra le entità PERSONA e CITTA'.

**Chiave:** E' uno speciale attributo, obbligatoriamente presente in ogni entità, che serve ad identificarne univocamente una specifica occorrenza.

Questo è l'unico attributo che DEVE necessariamente avere un valore: gli altri possono anche avere valore nullo.

Chiariamo anche questo concetto con un esempio, sempre riferito alla figura 2.

Nell'entità PERSONA possono esserci più individui con lo stesso nome e cognome, ma il *Codice Fiscale* sarà comunque diverso; esso diventa dunque chiave e graficamente lo si rappresenta con il pallino dell'attributo annerito. Sempre in riferimento alla figura 2, il fatto di indicare come chiave l'attributo *Nome* dell'entità CITTA' impone che non potranno esserci due (o più) città con lo stesso nome.

Il più delle volte, ed è anche il caso della nostra applicazione, però la chiave non è un attributo rappresentativo (come lo è il codice fiscale per una persona), bensì un numero naturale che si incrementa di una unità ad ogni inserimento di una nuova occorrenza. Questo tipo particolare di chiave è solitamente chiamato *Id*.

<sup>6</sup> Nel nostro caso avremo soltanto attributi su entità.

<sup>7</sup> Vedremo a breve perché *Codice Fiscale* e *Nome* (nell'entità CITTA') sono colorati di nero a differenza di *Cognome*

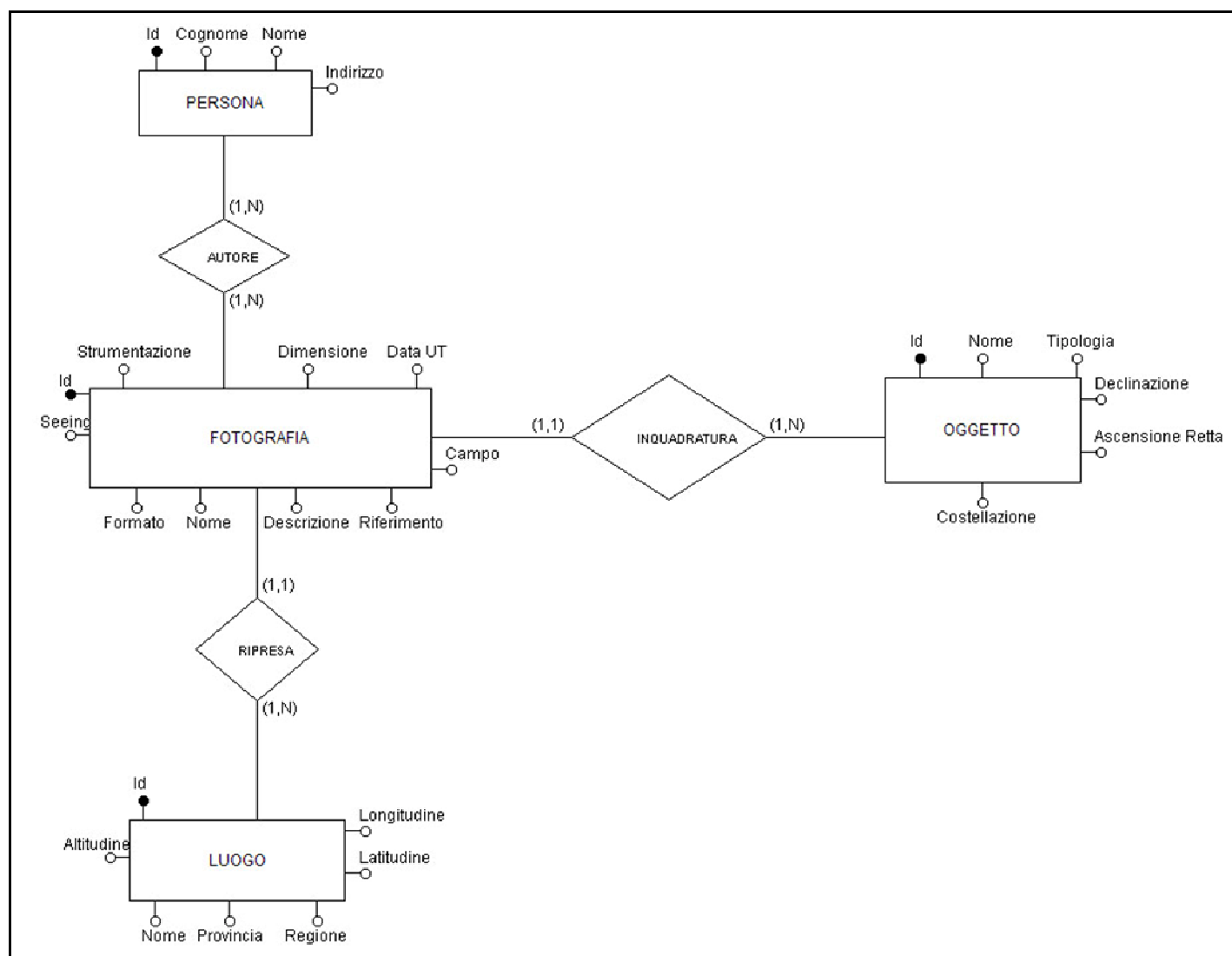


Figura 3. Il modello Entity Relationship per il database UAI

### Il modello Entity Relationship per il database UAI

A questo punto della trattazione non resta che presentare e commentare il modello *E-R* preliminare per il progetto UAI, figura 3. Come si può vedere lo schema è caratterizzato da quattro entità e tre relazioni.

Le entità sono FOTOGRAFIA, PERSONA, LUOGO e OGGETTO; le relazioni sono AUTORE, RIPRESA e INQUADRATURA.

Ogni entità ha come chiave un *Id*, secondo il concetto esposto nel paragrafo precedente. Come si può vedere anche graficamente l'entità che ricopre il ruolo centrale è FOTOGRAFIA: di questa fa parte ogni singola fotografia che si intende memorizzare nel sistema; grazie alla chiave incrementale possiamo pensare di memorizzare anche molte immagini dello stesso oggetto scattate nello stello istante ma in luoghi e da autori differenti, mantenendo assicurata la possibilità di discriminare tra una e l'altra.

Le altre entità sono tutte collegate a FOTOGRAFIA mediante le relazioni, mentre non vi sono collegamenti tra PERSONA, LUOGO e OGGETTO.

Vediamo nella tabella 1 più nei dettagli gli attributi dell'entità FOTOGRAFIA.

Tabella 1. Gli attributi dell'entità FOTOGRAFIA

Attributo	Descrizione
<b>Id</b>	Chiave unica di identificazione
<i>Nome</i>	E' un nome mnemonico riferito all'immagine, ad esempio se inquadrriamo M44 possiamo indicare "M44"
<i>Dimensione</i>	E' la dimensione espressa in Kbyte (o suoi multipli) della fotografia;
<i>Strumentazione</i>	Indica la strumentazione ottica e fotografica utilizzata per scattare la fotografia;
<i>Data UT</i>	Come è intuitivo, rappresenta la data e l'ora della ripresa, nel formato gg.mm.aaaa hh.mm.ss,sss

<i>Formato</i>	Indica il formato del file immagine: JPEG, RAW, FITS
<i>Riferimento</i>	E' il puntatore al file dell'immagine, solitamente memorizzata sugli <i>hard disk</i> del <i>server</i>
<i>Descrizione</i>	E' un campo piuttosto generico in cui si possono inserire informazioni non diversamente classificabili, ad esempio "Immagine scattata con seeing buono/discreto, ecc";
<i>Campo</i>	Rappresenta la dimensione (in secondi d'arco) del campo inquadrato;
<i>Seeing</i>	Indica la condizione di seeing della serata in cui è stata scattata la fotografia

L'entità PERSONA è pensata per memorizzare alcune informazioni base circa gli autori delle fotografie, il significato dei singoli attributi è evidente e non necessita di ulteriori commenti.

Anche il significato dell'entità LUOGO è piuttosto intuitivo: contiene le informazioni circa il luogo da cui è stato scattata l'immagine, con l'indicazione delle coordinate geografiche e l'altitudine (tabella 2).

**Tabella 2.** Gli attributi dell'entità LUOGO

Attributo	Descrizione
<i>Id</i>	Chiave unica di identificazione
<i>Nome</i>	Indica il nome del luogo: ad es. Colle del Nivolet
<i>Regione</i>	La regione italiana in cui è ubicato il luogo di osservazione; oppure lo stato estero
<i>Provincia</i>	La provincia in cui è ubicato il luogo di osservazione; oppure la località dello stato estero
<i>Longitudine</i>	Rappresentano le coordinate terrestri del luogo di osservazione
<i>Latitudine</i>	Vedasi sopra
<i>Altitudine</i>	E' l'altitudine sul livello del mare del luogo di osservazione

L'entità OGGETTO, invece, serve per memorizzare le informazioni circa l'oggetto astronomico inquadrato nella fotografia (se previsto); i suoi attributi descritti in tabella 3:

**Tabella 3.** Gli attributi dell'entità OGGETTO

Attributo	Descrizione
<i>Id</i>	Chiave unica di identificazione
<i>Nome</i>	E' il nome dell'oggetto: M45, Saturno, NCG xxxx, ecc
<i>Tipologia</i>	E' la tipologia dell'oggetto inquadrato (se prevista): Galassia, ammasso aperto, ammasso globulare, asteroide
<i>Costellazione</i>	La costellazione cui appartiene l'oggetto o il campo inquadrato
<i>Ascensione Retta</i>	Rappresentano le coordinate celesti inquadrate dall'immagine, più precisamente possiamo dire che rappresentano le coordinate del centro dell'immagine
<i>Declinazione</i>	Vedasi sopra

Procediamo ora a commentare le relazioni utilizzando anche le cardinalità.

AUTORE: Una persona deve essere autore di una o più fotografie (1,N); inoltre una fotografia può avere come autore una o più persone (1,N).

Come abbiamo visto prima, il significato dell'obbligatorietà della relazione è piuttosto intuitivo: se una persona non fosse autore di nessuna fotografia, non avrebbe senso memorizzarne i dati all'interno del database.

INQUADRATURA: Una fotografia deve inquadrare uno ed un solo oggetto (1,1); un oggetto può essere inquadrato in una o più fotografie (1,N).

In effetti questa clausola è piuttosto stringente: non è raro che all'interno dello stesso campo siano presenti due o più soggetti astronomici; pensiamo al caso di un'eclissi, in cui sia il soggetto occultante che quello occultato sono presenti contemporaneamente nella scena.

In questo contesto occorre specificare il soggetto principale andando poi ad utilizzare l'attributo *Descrizione* dell'entità FOTOGRAFIA per inserire le informazioni aggiuntive sul secondo (o terzo) oggetto inquadrato.

RIPRESA: Una fotografia deve essere ripresa da uno ed un solo luogo (1,1); da un luogo possono essere riprese una o più fotografie (1,N).

Questa struttura non troppo complessa, dovrebbe essere in grado di gestire la maggior parte delle situazioni che si presentano nell'uso quotidiano dell'astrofotografia.

Il passo successivo nel processo di ingegnerizzazione del *database*, che prende il nome di Progettazione Logica, consiste nella traduzione dello schema *E-R* nel modello Relazionale conseguente, vediamo nel paragrafo seguente.

### Il Modello Relazionale

Questo modello è stato introdotto nel 1970 da Codd [6] e a tutt'oggi la maggior parte dei sistemi di basi di dati commerciali e non presenti sul mercato, si basa su questa pubblicazione.

Non è compito di questo articolo descrivere in maniera esaustiva il modello, mi limiterò qui a definirne intuitivamente il concetto principale, rimandando nuovamente a [2] per la trattazione completa.

La *Tabella*, che rappresenta l'omologa struttura dell'entità nel modello *E-R*, è il cuore del modello Relazionale. Essa è esattamente come intuitivamente la si immagina, un esempio è riportato in tabella 4 in cui si riporta la tabella PERSONA.

**Tabella 4.** Un esempio di tabella nel modello Relazionale: la tabella PERSONA

<u>Id</u>	<u>Nome</u>	<u>Cognome</u>	<u>Indirizzo</u>
1	Pippo	Pluto	Via Roma 3
2	Paperino	Topolino	Via Pisa 1
3	.....	.....	.....

Come si vede nella prima riga (e scritti in grassetto) sono riportati i nomi dei campi (anche qui chiamati *attributi* come per il modello *E-R*) e nelle righe sottostanti i dati veri e propri, cioè il contenuto della tabella. In un *database* possono esserci molte tabelle ciascuna delle quali memorizza dati omogenei tra di loro<sup>8</sup>.

Schematicamente la tabella si indica in questo modo:

NOME\_TABELLA (Chiave<sub>1</sub>, Chiave<sub>2</sub>, ..., Chiave<sub>n</sub>,  
Attributo<sub>1</sub>, Attributo<sub>2</sub>, ..., Attributo<sub>n</sub>)

Indicando tra parentesi tonde l'elenco degli attributi, ricordandosi di sottolineare ed indicare per primi gli attributi chiave (l'equivalente del pallino annerito nel modello *E-R*)<sup>9</sup>, l'ordine degli altri attributi è, invece, irrilevante.

E' evidente che vi è un nesso molto stretto tra il modello *E-R* e quello Relazionale, infatti tramite un processo di

traduzione con regole formali è possibile passare dal modello concettuale (*E-R*), a quello logico (Relazionale), implementabile direttamente all'interno del *database server*.

Ecco quindi le tabelle risultanti dalla traduzione in Relazionale dello schema *E-R* di figura 3:

PERSONA (Id, Nome, Cognome, Indirizzo)

OGGETTO (Id, Nome, Tipologia, AscensioneRetta, Declinazione, Costellazione)

LUOGO (Id, Nome, Regione, Provincia, Longitudine, Latitudine, Altitudine)

AUTORE (IdPersona, IdFotografia)

FOTOGRAFIA (Id, Nome, Descrizione, Strumentazione, DataUT, Campo, Seeing, Formato, Dimensione, Riferimento, IdOggetto, IdLuogo)

Come si nota il processo di traduzione ha apportato due modifiche al modello *E-R*: la prima è la conversione in tabella della relazione AUTORE. Questo fatto è determinato dalle regole di traduzione che per una relazione N-N impone di introdurre una tabella nuova, "artificiale" per così dire.

D'altro canto le due relazioni INQUADRATURA e RIPRESA sono state condensate in due attributi "artificiali" inseriti nella tabella FOTOGRAFIA; essi sono *IdOggetto* ed *IdLuogo* che altro non sono se non le chiavi identificative rispettivamente per le entità OGGETTO e LUOGO; anche questa modifica deriva dalle regole di traduzione, il cui elenco completo, con le relative spiegazioni, è reperibile su [7].

Con questo ultimo passaggio abbiamo finalmente lo strumento con il quale si può implementare la struttura principale del *database UAI*.

### Conclusioni

In realtà non è affatto corretto intitolare questo paragrafo "conclusioni", perché di concluso in effetti c'è ancora poco.

In questo articolo mi sono occupato di descrivere un possibile modello per la base di dati del progetto UAI, restano ancora da definire le politiche di implementazione e le scelte della piattaforma tecnologica che dovrà ospitare il sistema una volta che sarà in produzione.

Trattandosi di una piattaforma web, consiglierai certamente un server *unix* (o *unix-like*) *Apache*<sup>10</sup>, dimensionato per qualche centinaio di accessi

<sup>8</sup> Abbiamo accennato che nello *Sloan Digital Sky Survey's* le tabelle sono addirittura centinaia.

<sup>9</sup> Questo modo di indicare le tabelle è una convenzione utilizzata in tutti i libri di testo.

<sup>10</sup> <http://www.apache.org/>

concorrenti, con pieno supporto al linguaggio *PHP*<sup>11</sup> e al database *MySQL*<sup>12</sup>.

Anche lo spazio di memorizzazione dovrà essere dimensionato attentamente, in modo da poter contenere una gran quantità di immagini. Questo calcolo sarà possibile non appena saranno definiti i formati delle immagini: risoluzione, dimensione, formato-colore, ecc.

E' da valutare l'opportunità di iniziare lo sviluppo del sistema ex-novo o personalizzare ad hoc un *CMS* già esistente; la scelta probabilmente dipenderà da come sarà dimensionata la squadra di programmatori che lavorerà sul sistema.

Eventualmente altre considerazioni su questo tema verranno esposte in un articolo successivo, dopo che saranno recepite le indicazioni emerse in sede congressuale.

Gli argomenti sono ancora molti e vari, ma aver stabilito i punti fermi sul procedimento di definizione della struttura della base di dati rappresenta il punto di partenza per ogni altra azione che la comunità deciderà di intraprendere.

banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377-387, 1970.

- [7] P. Atzeni, S. Ceri, S. Paraboschi e R. Torlone, «La progettazione logica,» in *Basi di dati - Modelli e linguaggi di interrogazione*, Milano, McGraw-Hill, 2009, pp. 311-322.

## Bibliografia

- [1] F. Pasian and R. Smareglia, "WWW Access to Astronomical Archives and Databases," *International Journal of Modern Physics C*, pp. 817-830, 22 Giugno 1994.
- [2] P. Atzeni, S. Ceri, S. Paraboschi e R. Torlone, «Il modello relazionale,» in *Basi di dati, Modelli e linguaggi di interrogazione*, Milano, McGraw-Hill, 2009, pp. 17-42.
- [3] A. R. Thakar, A. Szalay, G. Fekete and J. Gray, "The Catalog Archive Server Database Management System," *Computing in Science & Engineering*, pp. 30-37, Gennaio, Febbraio 2008.
- [4] P. Atzeni, S. Ceri, S. Paraboschi e R. Torlone, «Metodologie e modelli per il progetto,» in *Basi di dati Modelli e linguaggi di interrogazione*, Milano, McGraw-Hill, 2009, pp. 207-244.
- [5] P. Chen, "The Entity-Relationship model: Toward a unified view of data.," *ACM Transaction on Database System*, vol. 1, no. 1, pp. 9-36, 1976.
- [6] E. Codd, "A relational model for large shared data

---

<sup>11</sup> <http://www.php.net/>

<sup>12</sup> <http://www.mysql.it/>